



#9
1/7/04
Roy

Patent
Attorney's Docket No. 032885-001

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of)

John P. WONG)

Application No.: 09/810,246)

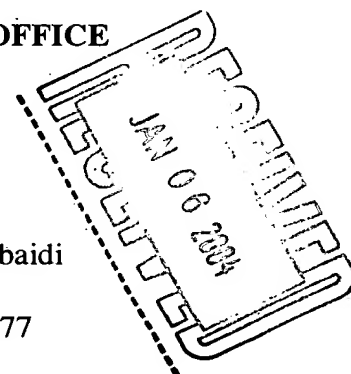
Filed: March 19, 2001)

For: MIRROR FILE SYSTEM)

Group Art Unit: 2171

Examiner: H. J. Alaubaidi

Confirmation No.: 8377



REQUEST FOR RECONSIDERATION

RECEIVED

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

JAN 05 2004
Technology Center 2100

Sir:

In response to the Office Action dated October 2, 2003, Applicant respectfully requests reconsideration and withdrawal of the rejections of the claims. The indication that claims 13 and 16 contain allowable subject matter is noted with appreciation. The rewriting of those claims in independent form is being held in abeyance, pending consideration of this response.

Claims 1-10 were rejected under 35 U.S.C. 103(a) as being unpatentable over the Kramer patent (U.S. Patent No. 6,216,140) in view of the Skiba patent (U.S. Patent No. 6,366,5988). Claims 11, 12, 14 and 15 were rejected under 35 U.S.C. 103(a) as being unpatentable over the Skiba patent in view of the Kramer patent. Applicant respectfully submits that these patents do not disclose, nor otherwise suggest, the subject matter of the rejected claims, whether considered individually or in combination.

General Overview

The Kramer patent deals with a particular way of using files and directories, which are *elements* of a file system. It does not teach how to manipulate or control any file system as a whole, whether local or remote. This puts the patent into sharp contrast with the presently claimed invention. Kramer uses some *elements* of a file system; whereas the claims are directed to a *new type* of file system.

The Skiba patent deals with storage, or volumes, of a single local file system on a single host. Its Redirector, Migrator, Versioning, and Mirroring work only on storage and volumes under a local file system. Skiba does not deal at all with the file system itself, nor with the essential operations involved in mounting, unmounting, or accessing the file system as a whole.

The Mirror File System of the claimed invention deals with the following:

- a. How to construct a "super" or "meta" file system, namely the Mirror File System (MFS), on top of two or more existing files systems, whether local or remote. (Figs. 2, 5, 6);
- b. How to manage two or more file systems through a single mount point, using a unique mounting mechanism. (Page 17, Lines 19-21);
- c. How to link two file systems as a mirroring pair, such that a write operation from an application sends or "writes" to both file systems simultaneously in real time. (Page 21, Lines 2-4; Page 23, Lines 28-32);
- d. How to use the Mirror File System to mirror files between two file systems on *two separate network servers* in real time. (Page 27, Line 17 - Page 28, Line 8);
- e. How to unmount, or "break", the mirroring using a unique unmount protocol. (Page 20, Lines 21-28);
- f. How to provide complete transparency to applications via MFS's VFS structure (called *mfs_vfs*) and its Vnode operation structure (called *mnode*). (Page 16, Lines 8-24); and

g. How to provide data coherency and consistency by means of a unique locking protocol (Page 24, Line 20 - Page 25, Line 6).

Discussion of Rejections

In rejecting claims 1-10, the Office Action states:

Regarding Claims 1-3 and 7-9, Kramer discloses:
mounting components of each of said two physical file systems in a single directory
(Col 3, Lines 16-19), i.e.

There is therefore an unmet need in the art to be able to perform
a merge operation of two or more hierarchies of files and directories
that merges content as well as attribute differences in an efficient
manner.

(Col 3, Lines 35-38), i.e.

It is still another object of the invention to be able to perform a
merger operation of two or more hierarchies of files and directories
that merges content as well as attribute differences in an efficient
manner.

(Col 9, Lines 51-54), i.e.

the merge operation typically occurs one item at a time, thereby being
an exceedingly time-consuming process especially for every hierarchy
of files and directories.

(Col 13, Lines 38-40), i.e.

merging the first item of the first hierarchy with the second item of
the second hierarchy into a single entry in the difference list

Rejected claims 1-3 and 7-9 concern:

1. How the MFS mounts two physical file systems in a single directory.
2. How MFS manages these two physical file systems after the mount operation has taken place, by means of the virtual file system (MFS) data structure.
3. The fact that the MFS data structure contains elements which correspond respectively to each of the mounted components (i.e., to each of the mounted physical file systems).

The Kramer patent discloses the merger of two or more hierarchies of files and directories. This merge operation operates on two hierarchies *inside a single file system*. It does *not* involve:

- * mount operations on *physical file systems*, nor
- * any data structure containing information about *mounted physical file systems*.

It is respectfully submitted that Kramer's disclosure has no relevance or similarity to the rejected claims.

The Office Action goes on to state:

A virtual file system data structure containing elements which respectively correspond to each of the mounted components (Col 3, Lines 39-45), i.e.

Therefore, according to a first aspect of the present invention, efficient copying and sharing of large amounts of hierarchically organized information is provided by a method that **first creates a virtual copy of a hierarchy of items of files and directories by adding a new link to the root of the hierarchy in order that the hierarchy may be shared by one or more versions of code.**

Kramer's "virtual copy and adding a new link" pertains only to files and directories as components of a single file system. This functionality operates only on these elements (files and directories), not on an entire file system, and certainly not on *multiple file systems*.

The claimed Mirror File System, on the other hand, uses its virtual file system data structure to contain information about *two or more mounted physical file systems*; it further uses this "meta" information to *manage* the mounted physical file systems. This functionality operates on *entire file systems*, not merely on particular elements or components. In the world of data storage and management, this is a significant distinction. Although Kramer's disclosure may be seen to operate on "large amounts of hierarchically organized information", this phrase is somewhat ambiguous. Generally speaking, the

functionality of the claimed invention operates on quantities of information and organizing structures that are at least an order of magnitude larger than Kramer's.

Continuing on, the Office Action states:

Kramer reference discloses all of the claimed subject matter set forth above, except it does not explicitly indicate the step of each of said elements having an application interface data structure with two associated pointers that respectively point to application interface data structure of a corresponding component in each of said two physical file system. However Skiba teaches wherein with two associate pointers that respectively point to application interface data structure of a corresponding component in each of said two physical file system (Col 2, Lines 48-54), i.e.

Another technique for creating an expanded volume was the use of a Distributed File System such as Microsoft's DFS. Using DFS software, a logical volume can be created where each subfolder may point to a folder on a different volume on any machine in the network. This approach is very useful for creating logical directory structure independent of volume location. Given the intended broad application of the Kramer system, it would have been obvious to a person of ordinary skill in the art at the time of Applicant's invention to modify the teachings of Kramer with the teachings of Skiba to include pointers to components in other file system and not just the current file system to increase the system performance by allowing access to a second file system, which is better than just one file system especially in network traffic congestion, one other reason would be to allow access to other file systems for recovery and backup situation.

The claimed invention is directed to a *completely new file system*, the Mirror File System, which, by enhancing a basic VFS file system application interface structure, mounts two physical file systems on a single mount point and links, mirrors, and manages these mounted physical file systems. From a technological point of view, this is a completely different approach from simply adding two pointers into the basic file system data structure; it operates on a much larger scale, produces different and more significant effects, and is based on different software architecture and programming techniques.

Skiba's scheme creates an expanded volume under Microsoft's DFS. Creating an expanded volume does not involve using a basic VFS file system interface structure, does not perform mounting operations, and does not involve other physical file systems. Skiba's

disclosure has no relevance or similarity to the unique mounting operation of the claimed invention.

The Office Action next states:

Regarding Claims 4 and 10, the first limitation of this claim has been noted in the rejected claim 1, above. In addition Kramer teaches receiving a request to perform a write operation (figure 2B, Element No. 22, see also Col 15, Lines 47-50)

performing said write operation on both copies of said one component in said two physical file systems in real time (Col 15, Lines 43-46), i.e.

In the current invention, mirroring means that all modifications in the original file are identically applied to the mirror copy as soon as these modifications are made. (Col 15, Lines 51-53), i.e.

Generally, mirroring is done on each Write request by writing input data to the mirror copy in addition to default processing

The mirroring functionality disclosed by Skiba pertains to mirroring a file between two storage volumes managed by a single file system, with a second copy of a file used for backup and recovery purposes only, and accessible only through the original file system. The claimed Mirror File System, on the other hand, mirrors a file between two *different* and *separate* physical file systems. Typically, one is a local file system; another is located remotely on a different server. The difference in functionality and effect is that under the MFS, the second copy of a file can be accessed *immediately* by a second file system on another remote server.

In addition, the methods of writing these two copies are different. Skiba's mirroring is done on the device driver level; the MFS operates on the file system level. MFS mirroring requires a locking mechanism, recited in claim 6, to ensure data coherency and consistency. Skiba's mirroring a file between different volumes does not utilize the locking mechanism and does not ensure data coherency or consistency.

The Office Action next states:

Regarding Claim 5, Skiba discloses path name(Col 16, Lines 30-32). i.e.

FIG. 13 represents the scheme of mirroring files under the Windows NT platform. The file request comes from the I/O Manger 42 and contains the full file name.

Skiba's full file name is the access point for two copies on two volumes under a *single file system*, whereas the MFS uses a path name as the access point for two copies on *two different physical file systems*.

The Office Action goes on to state:

Regarding Claim 6, Skiba discloses acquiring a lock (Col 15, Lines 47-50), i.e.

The Mirroring Driver processes numerous file requests, including but not limited to: Open, Write(Read for special case), Lock/Unlock, Close, Delete, and Rename, which is used to handle file name and location synchronization.

Skiba's disclosure indicates that a file operation including Lock/Unlock must be provided by the single file system that can be processed by the mirroring driver. However, Skiba's disclosure has nothing to do with techniques or methods for acquiring a lock or semaphore on the behalf of a Mirroring driver. The rejected claim pertains to how the Mirror File System achieves data coherency and consistency between the copies of a file that reside on two different physical file systems. It uses a unique locking protocol to avoid the possibility of a deadlock situation.

In connection with the rejection of claims 11, 12, 14 and 15, the Office Action states:

Regarding Claims 11-12 and 14-15, Skiba discloses a first and second server having a first and a second local file system (Col 13, Lines 59-62; see also Col 17, Lines 5-12)

Skiba reference discloses all of the claimed subject matter set forth above, except it does not explicitly indicate the step of client device having a virtual file system which mounts an imported file system from both file systems.

The lines in Col. 13 and Col. 17 cited in the Office Action appear to be unrelated to the server and file system. In addition, Skiba proposes creation of an expanded volume for a single file system to be used for backup and recovery purposes. The reference has nothing to do with a virtual file system, which mounts an imported file system.

The Office Action next states:

Kramer discloses the step of client device having a virtual file system (Col 3, Lines 39-45) which mounts an imported file system from both file systems

(Col 3, Lines 16-19), i.e.

There is therefore an unmet need in the art to be able to perform a merge operation of two or more hierarchies of files and directories that merges content as well as attribute differences in an efficient manner.

(Col 3, Lines 35-38), i.e.

It is still another object of the invention to be able to perform a merge operation of two or more hierarchies of files and directories that merge content as well as attribute difference in an efficient manner.

(Col 9, Lines 51-54), i.e.

the merge operation typically occurs one item at a time, thereby being an exceedingly time-consuming process especially for very large hierarchies of files and directories.

(Col 13, Lines 38-40), i.e.

merging the first item of the first hierarchy with the second item of the second hierarchy into a single entry in the different list

Kramer proposes a merge operation on two hierarchies of files/directories, whereas the Mirror File System mounts imported files from remote servers. Although some file systems are hierarchical, the phrase "two hierarchies of files/directories" is not at all synonymous with two separate physical file systems. Kramer's operational functionality is much narrower in scope and pertains to much smaller organizational units and structures than the claimed subject matter.

The Office Action then states:

Given the intended broad application of Skiba's system, it would have been obvious to a person of ordinary skill in the art at the time of applicant's invention to

modify the teachings of Skiba with the teachings of Kramer to include client device having a virtual file system, which mounts an imported file system from both servers to prevent the system from losing any data by providing multiple servers instead of storing all the data on one server.

As noted above, Skiba's system creates an expanded volume for a single file system, and Kramer's system uses files or directories to do versioning control. Neither mounts two file systems under a single directory to form a mirroring pair. They do not operate at the file system level, and their disclosures have no bearing on the invention or design of a file system. Further, although certain ideas may seem obvious in retrospect, it is respectfully submitted that only the present application discloses how to design and implement the technology encompassed by the Mirror File System. In any case, neither Kramer nor Skiba deals effectively with the problems addressed nor the solutions proffered by the present invention, nor have they implemented or suggested solutions similar to the invention.

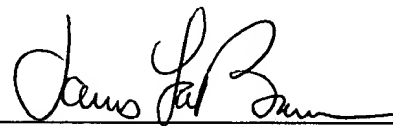
Reconsideration and withdrawal of the rejections, and allowance of all pending claims, are therefore respectfully requested.

Respectfully submitted,

BURNS, DOANE, SWECKER & MATHIS, L.L.P.

Date: January 2, 2004

By: _____



James A. LaBarre

Registration No. 28,632

P.O. Box 1404
Alexandria, Virginia 22313-1404
(703) 836-6620